COMMENTS

The enclosed is responsive to the Notice of Non-Compliant Amendment Office Action mailed on November 14, 2007. At the time the Examiner mailed the Office Action claims 1-36 were pending. By way of the present response the Applicant has: 1) amended claims 1, 17 -18, 26 and 34; 2) not canceled any claims; and, 3) not added any claims. As such, claims 1-36 remain pending. The Applicant respectfully requests reconsideration of the present application and the allowance of claims 1-36.

35 USC 101 Rejections

The Examiner has rejected independent claims 1 and 18 under 35 USC 101 as being directed to non-statutory subject matter. According to the Examiner, "[t]he claims appear to be define the metes and bounds of an invention comprised of software alone without claiming associated computer hardware required for execution. Software alone, without a machine, is incapable of transforming any physical subject matter by chemical, electrical, or mechanical acts. Although the Applicant respectfully disagrees with the Examiner, in order to advance the prosecution of the present application towards allowance, the Applicant has amended the respective preambles of claims 1 and 18 to recite "[a]n integrated logging and tracing system employed within a system of computers interconnected through a network." The Applicant respectfully submits that claims 1 and 18 are directed to patentable subject matter at least in view of the above described amendment.

The Examiner has rejected claim 34 as being directed to non-statutory subject matter. According to the Examiner, the Applicant's "specification defines several articles of manufacture including non-statutory media such as data signals embodied in a carrier

Application No.: 10/749,616   - 10/16-   06570P024/2003P00502
Amdt. dated 12-14-2007
Reply to Notice of Non-Compliance of 11/14/2007

wave . . . ". The Applicant respectfully disagrees with the Examiner. Without addressing the issue of whether or not signals embodied on a carrier wave are patentable or not, the Applicant respectfully submits that claim 34 does not contemplate carrier waves or other propagation media because, as originally written, claim 34 recites "[a]n article of manufacture having program code <u>stored thereon</u> . . .". See, Examiner's Office Action, mailed 7/6/07, p. 2. Carrier waves and propagation media are not capable of storing program code. Therefore claim 34 as written does not offend the Examiner's definition of patentable subject matter.

35 USC 103 Rejections

The Examiner has rejected independent claims 1, 18, 26 and 34 under 35 USC 103 as being obvious in view of the combination of U.S. Pat. No. 6,539,501 (hereinafter, "Edwards") and U.S. Pat. No. 7,174,370 (hereinafter, "Saini"). The Examiner has reasoned that Edwards teaches both a tracing module and a logging module but fails to teach the presence of a network and that Saini teaches the missing network. See, Examiner's Office Action, 07/06/07, p. 3.

The Edwards reference teaches a system in which application software code is instrumented with monitoring code, referred to as a "log method" that interfaces with a "logger daemon" (12) in order to fully perform application monitoring. The log method writes (or causes the logger daemon to write) the information being traced into a trace queue (16). The logger daemon empties the queue by writing the trace data into a more permanent storage means (the "log file"). The following citations from Edwards are noteworthy in this regard.

In preferred embodiments, the user would insert one or more <u>log methods</u> into the source code 6 <u>as trace statements to monitor the execution path of the source code</u>.

Col. 3, lines 16-18.

During execution, execution of the <u>log method</u> will cause the thread executing the object code 10 to generate trace data and then call the <u>logger daemon</u> to further handle the trace data.

Col. 3, lines 58-61.

In preferred embodiments, the thread executing the object code 10 would buffer gathered trace data into <u>the trace queue 16</u> and the <u>logger [daemon] 12</u> would write the trace statements to <u>the log file 14</u>.

Col. 3, lines 8-13.

If the developer decides to activate the tracing function, i.e., TraceOn . . . then a separate <u>logger daemon 12</u> executing independently of the thread executing the object code will spawn a thread to monitor <u>the trace queue 16</u> and write trace statements to <u>the log file 14</u>.

Col. 5, lines 12-17.

The above statements indicate that the Edwards reference describes a system in which a "log method" and a "logger daemon" work together to perform 'tracing'. The following statement of Edwards also indicates that, through the use of the unique ID, the "log method" and "logger daemon" may even work together to perform 'logging'.

A unique ID 20 provides an identifier for a group of which the log method is associated. For instance, a Java applet program may be comprised of different components, each performing some particular defined function <u>or category</u> of functions within the applet. In such a case, the software developer may assign a unique ID to each component of the applet , such that the log methods having the unique ID associated with a particular component would be used to trace the execution of those Java statements related to that particular component.

Col. 3, lines 19 - 28.

However, the Applicant respectfully submits that the teachings of the Edmonds reference differ from those of the present application at least in the following respect: whereas the log method and logger daemon of Edmonds appear to be completely different functional entities having little or no overlap of functions between them (the log method causes traces to be written into the queue while the logger method empties the queue), by

contrast, the tracing and logging modules of the present application are 'different but similar' having some functional overlap between them but not being identical. The 'different but similar' aspect of the tracing and logging modules manifests itself from their being instances of different sub-classes of the same class. That is, they <u>inherit</u> 'same' properties from the primary class but <u>inherit</u> 'different' properties from their respective sub-classes of the primary class.

The Applicant has amended independent claims 1 and 18 in order to emphasize this distinction between the teachings of the Edmonds and the present application. Independent claims 26 and 34 are already included this distinction and so no amendment has been made in these claims.

With respect to the amendments made to claims 1 and 18, independent claim 1 has been amended as follows:

> 1. (Currently Amended) An integrated tracing and logging system employed within <u>a system of computers interconnected through</u> a network comprising:
> a tracing module associated with specified program code regions of an application, the tracing module to receive and process tracing method calls generated by the application when the specified program code regions are executed<u>, the tracing module being an instance of a first sub-class of a class</u>; and
> a logging module associated with specified categories <u>of the system</u> ~~related to the network~~, the logging module to receive and process logging method calls from ~~network~~ components associated with the categories<u>, the logging module being an instance of a second sub-class of the class, the first sub-class being different than the second sub-class.</u>

With respect to the amendment appearing in the preamble ("a system of computers interconnected through a network"), the Applicant respectfully submits that support for this amendment can be found in paragraphs [0002] (which discusses generally the notion of logging into a server connected to a network; and [0010] (indicating that logging modules receive and process method calls from network components). Here, the Applicant notes that the logging function alone is sufficient to disclose "a system of computers interconnected through a network".

With respect to the amendments made to the body of the claim the Applicant apologizes to the Examiner for having not only failed to identify the portions of the specification that support these amendments but also for making a mistake in the amendment itself. Specifically, comparing the amendment of 8/23 with the present amendment, please note that the applicant has changed the second, added recitation of "tracing module" to recite "logging module" instead.

Support for the amendment that "the tracing module [is] an instance of a first sub-class of a class" and "the logging module [is] an instance of a second sub-class of the class, the first sub-class being different than the second sub-class" is found in paragraph [0020]. Paragraph [0020] recites in part (emphasis added):

[0020] In an embodiment, each controller 200 is an instance of a defined "controller" class (e.g., a Java class) which includes two sub-classes, a "tracing" sub-class and a "logging" sub-class (described in detail below), which provide features specific to tracing and logging operations, respectively. In an object-oriented environment such as Java, the tracing controller 202 illustrated in Figure 2 represents an instance of the tracing sub-class and the logging controller 204 represents an instance of the logging sub-class . . . .

The Applicant respectfully submits that the above description clearly supports the claim amendment at issue. One of ordinary skill in the art would clearly understand the above paragraph to be referring to object-oriented technology a basic property of which is the notion of "inheritance" in which a child object/class inherits the properties of its parent object/class.

For instance, as a simple example, a generic "car" class might define generic properties of a car (e.g., i) 6 cylinder, and, ii) sedan) while sub-classes of the generic car class might define additional specifics of these properties for a particular make and model of the car. For example, a "Saturn" car sub-class object might include i) 6 cylinder, dual overhead CAM; and ii) sedan color = black, while, a "Hudson Hornet" car sub class object might include i) 6 cylinder, twin carburetor; ii) sedan color = purple. Both of the sub-classes have some overlap (6 cylinder and a sedan) but some difference too (dual

Application No.: 10/749,616   - 14/16-   06570P024/2003P00502
Amdt. dated 12-14-2007
Reply to Notice of Non-Compliance of 11/14/2007

overhead CAM vs. twin carburetor and black vs. purple). One of ordinary skill in reading paragraph [0020] of the instant application would immediately understand the existence of such a relationship as between the logging and tracing sub-classes. Both are children of the controller class and would therefore inherit properties of the controller class, but, both are different sub-classes and therefore have differences between them.

The other amendments are supported for similar reasons. The Applicants respectfully submit that all claim amendments are supported by the specification as explained above.

Closing Comments

Because the Applicant has demonstrated the patentability of all pending independent claims, the Applicant respectfully submits that all pending claims are allowable. The Applicant's silence with respect to the dependent claims should not be construed as an admission by the Applicant that the Applicant is complicit with the Examiner's rejection of these claims. Because the Applicant has demonstrated the patentability of the independent claims, the Applicant need not substantively address the theories of rejection applied to the dependent claims. Moreover, where the Applicant has failed to address a specific independent claim element alleged by the Examiner to be covered by prior art, such failure should not be viewed as an admission by the Applicant that the Applicant accepts or agrees with the Examiner's reasoning.

In the further interests of efficiency, the Applicant reserves the right under MPEP 2144.03.C to cause the Examiner to find in the prior art subject matter to which the Examiner has taken Official Notice at a later time in the prosecution of the present case when the subject matter of such prior art is actually at issue.

CONCLUSION

Applicant respectfully submits that the present application is in condition for allowance. If the Examiner believes a telephone conference would expedite or assist in the allowance of the present application, the Examiner is invited to call Robert B. O'Rourke at (408) 720-8300.

Pursuant to 37 C.F.R. 1.136(a)(3), applicant(s) hereby request and authorize the U.S. Patent and Trademark Office to (1) treat any concurrent or future reply that requires a petition for extension of time as incorporating a petition for extension of time for the appropriate length of time and (2) charge all required fees, including extension of time fees and fees under 37 C.F.R. 1.16 and 1.17, to Deposit Account No. 02-2666.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Date: _12|14_, 2007

Robert B. O'Rourke
Reg. No. 46,972

1279 Oakmead Parkway
Sunnyvale, CA 94085-4040
(408) 720-8300